

PIXBAY Gallery

This Flutter application is an image gallery that fetches and displays images from the **Pixabay API**. The images are displayed in a responsive grid layout, where the number of columns adjusts based on the screen width. Users can scroll down to load more images automatically. Each image shows the number of likes and views underneath it. This application uses the http package to fetch data from the API and the **cached_network_image** package to efficiently load and cache images.

Flutter

Flutter is an open-source UI toolkit developed by Google for building natively compiled applications across multiple platforms from a single codebase. It enables developers to create visually appealing and performant apps for mobile, web, and desktop platforms using the Dart programming language. Flutter provides a rich set of pre-designed widgets and tools that help in crafting complex and responsive UIs. It also boasts features like hot reload, which speeds up the development process by allowing developers to see code changes in real-time. Due to its versatility and growing community support, Flutter has become a popular choice for cross-platform development.

Pixabay API

The Pixabay API is a web-based service that allows developers to programmatically access and retrieve a large collection of high-quality, royalty-free images and videos. Pixabay is a popular online platform where users can find media content for personal and commercial use without the need for attribution. By integrating the Pixabay API, developers can easily search for and display media content in their applications. The API offers various search filters such as image type, category, and orientation, making it easy to customize content retrieval based on specific needs.

pub.dev

pub.dev is the official package repository for the Dart and Flutter ecosystems, where developers can discover, use, and share packages. Packages on pub.dev are pre-built libraries or tools that extend the capabilities of Flutter and Dart, enabling developers to add functionality such as animations, database access, HTTP requests, and more without having to write everything from scratch. The repository provides detailed documentation, version management, and community ratings, making it a valuable resource for the Flutter community. By leveraging packages from pub.dev, developers can save time and streamline the development process.

Packages Used in This Project

This project utilizes a few essential Flutter packages to build the image gallery application. The `http` package is used to make HTTP requests to the Pixabay API and fetch image data. It simplifies the process of sending requests and parsing responses. The `cached_network_image` package is employed to efficiently load and cache images fetched from the API, improving the app's performance and reducing data usage by caching images locally. Additionally, the `intl` package can be useful for formatting numbers if needed. These packages collectively enhance the app's functionality, making it efficient and user-friendly.

Flutter Codes

Main.dart

```
import 'package:flutter/material.dart';
import 'package:pixbay_gallery/view/home/home.dart';

void main() {
  runApp(const PixaBayGalleryApp());
}

class PixaBayGalleryApp extends StatelessWidget {
  const PixaBayGalleryApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Pixabay Gallery',
      debugShowCheckedModeBanner: false,
      home: ImageGalleryScreen(),
    );
  }
}
```

Pixabay API Function

```
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:pixbay_gallery/controller/services/APIsNKeys.dart';

class PixabayApi {
  int page = 1;
```

```
PixabayApi();
```

```
Future<List<dynamic>> fetchImages() async {  
  final url = Uri.parse(  
    'https://pixabay.com/api/?key=$pixabayAPIKey&image_type=photo&per_page=20&page=$page');  
  final response = await http.get(url);  
  
  if (response.statusCode == 200) {  
    final data = json.decode(response.body);  
    page++;  
    return data['hits'];  
  } else {  
    throw Exception('Failed to load images');  
  }  
}
```

Home.dart

```
import 'package:cached_network_image/cached_network_image.dart';  
import 'package:flutter/material.dart';  
import 'package:pixbay_gallery/constants/commonFunctions.dart';  
import 'package:pixbay_gallery/constants/constants.dart';  
import 'package:pixbay_gallery/controller/services/pixabayAPIs.dart';
```

```
class ImageGalleryScreen extends StatefulWidget {  
  @override  
  _ImageGalleryScreenState createState() => _ImageGalleryScreenState();  
}
```

```
class _ImageGalleryScreenState extends State<ImageGalleryScreen> {  
  final PixabayApi api = PixabayApi();  
  final ScrollController _scrollController = ScrollController();  
  List<dynamic> images = [];  
  bool _isLoading = false;
```

```
@override  
void initState() {  
  super.initState();  
  _fetchImages();  
  _scrollController.addListener(() {  
    if (_scrollController.position.pixels ==  
        _scrollController.position.maxScrollExtent &&  
        !_isLoading) {
```

```

    _fetchImages();
  }
});
}

```

```

Future<void> _fetchImages() async {
  setState(() {
    _isLoading = true;
  });
  try {
    final newImages = await api.fetchImages();
    setState(() {
      images.addAll(newImages);
    });
  } catch (e) {
    print(e);
  } finally {
    setState(() {
      _isLoading = false;
    });
  }
}

```

```

@override
Widget build(BuildContext context) {
  final crossAxisCount = MediaQuery.of(context).size.width ~/ 150;

  return Scaffold(
    appBar: AppBar(
      title: const Text(
        'Pixabay Gallery',
        style: TextStyle(
          fontWeight: FontWeight.bold,
          fontSize: 30,
        ),
      ),
    ),
    body: GridView.builder(
      padding: EdgeInsets.symmetric(
        horizontal: MediaQuery.of(context).size.width * 0.03,
      ),
      controller: _scrollController,
      gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: crossAxisCount,

```

```

mainAxisSpacing: 15,
crossAxisSpacing: 10,
childAspectRatio: 0.8,
),
itemCount: images.length,
itemBuilder: (context, index) {
  final image = images[index];
  return Column(
    children: [
      Expanded(
        child: CachedNetworkImage(
          imageUrl: image['webformatURL'],
          // imageBuilder: (context, imageProvider) => Container(
          //   decoration: BoxDecoration(
          //     borderRadius: BorderRadius.circular(10),
          //   ),
          // ),
          placeholder: (context, url) =>
            const Center(child: CircularProgressIndicator()),
          errorWidget: (context, url, error) => const Icon(Icons.error),
          fit: BoxFit.cover,
        ),
      ),
    ],
  ),
  const SizedBox(height: 4),
  Row(
    children: [
      SizedBox(
        width: MediaQuery.of(context).size.width * 0.01,
      ),
      const Icon(
        Icons.favorite_outlined,
        color: Colors.red,
      ),
      Text(
        CommonFunctions.getShortForm(
          image['likes'],
        ),
        style: const TextStyle(
          fontSize: 15,
          fontWeight: FontWeight.w600,
        ),
      ),
    ],
  ),
  const Spacer(),
  const Icon(

```

```
Icons.remove_red_eye,  
color: Colors.black,  
)  
Text(  
  CommonFunctions.getShortForm(  
    image['views'],  
  ),  
  style: const TextStyle(  
    fontSize: 15,  
    fontWeight: FontWeight.w600,  
  ),  
)  
  SizedBox(  
    width: MediaQuery.of(context).size.width * 0.01,  
  )  
],  
)  
],  
);  
},  
)  
);  
}  
}
```